

11/12/1

FIG. 1

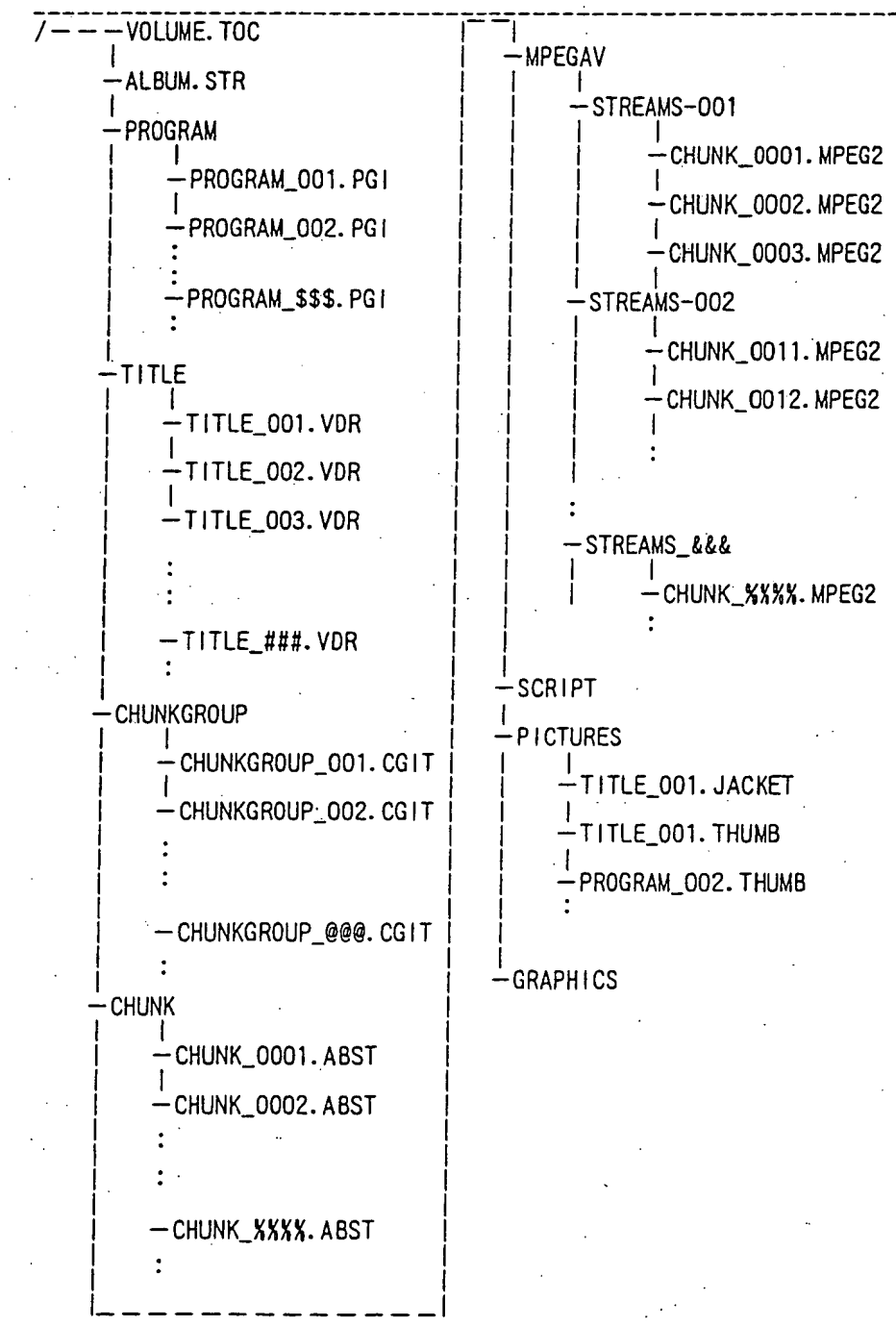


FIG. 2

Syntax	Number of Bits	Mnemonic
VOLUME.TOC{ file_type_id volume_information() text_block() }	8*16	char[16]

FIG. 3

Syntax	Number of Bits	Mnemonic
volume_information() { volume_attribute() resume() volume_rating() write_protect() play_protect() recording_timer() }		

FIG. 4

Syntax	Number of Bits	Mnemonic
volume_attribute() { volume_attribute_length vdr_version reserved title_playback_mode_flag program_playback_mode_flag volume_play_time() update_time_count() maker_id model_code POSID }	32 4*4 6 1 1 4*8 32 8*16 8*16 32	uimsbf bcd bslbf bslbf bslbf bcd uimsbf char[16] char[16] bslbf

F I G . 5

Syntax	Number of Bits	Mnemonic
resume() {		
resume_length	32	uimsbf
reserved // for byte alignment	3	bslbf
resume_switch	1	bit
reserved	4	bslbf
number_of_records	4	uimsbf
reserved // for byte alignment	7	bslbf
resume_auto_execute_time_flag	1	bit
resume_auto_execute_time()	4*14	bcd
reserved	4	bslbf
resume_auto_execute_record_number	4	uimsbf
for(i=0;i<number_of_records;i++) {		
resume_mode_flag	4	bslbf
object_type	4	bslbf
linked_record_number	4	uimsbf
number_of_times	16	uimsbf
resume_updated_time()	4*14	bcd
switch(object_type) {		
case title:		
title_number	16	uimsbf
title_local_time_stamp	64	uimsbf
break;		
case program:		
program_number	16	uimsbf
program_local_time_stamp	64	uimsbf
break;		
case program_bind:		
program_bind_number	16	uimsbf
program_order	16	uimsbf
program_number	16	uimsbf
program_local_time_stamp	64	uimsbf
break;		
case play_item:		
play_item_number	16	uimsbf
play_item_local_time_stamp	64	uimsbf
break		
}		
}		
}		

FIG. 6

Syntax	Number of Bits	Mnemonic
volume_rating() {		
volume_rating_id	8*16	char[16]
volume_rating_length	32	uimsbf
reserved	6	bslbf
volume_rating_type	2	bslbf
volume_rating_password	128	bslbf
switch (volume_rating_type) {		
case age_limited:		
number_of_rating	8	uimsbf
for (i=0; i<number_of_rating; i++) {		
country_code_for_rating	24	bslbf
age_for_volume_rating	8	uimsbf
}		
break;		
case CARA:		
reserved	4	bslbf
CARA_category	4	bslbf
reserved	24	bslbf
break;		
case RSAC:		
reserved	4	bslbf
RSAC_category	4	bslbf
reserved	4	bslbf
RSAC_level	4	bslbf
reserved	16	bslbf
break;		
}		
}		

FIG. 7

Syntax	Number of Bits	Mnemonic
write_protect() {		
write_protect_length	32	uimsbf
volume_write_protect_level	4	uimsbf
password_enable_flag	1	bslbf
append_only_flag	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_write_protect	128	bslbf
reserved	8	bslbf
write_protect_set_time()	56	bcd
reserved	8	bslbf
write_protect_expiration_time()	56	bcd
number_of_times	16	uimsbf
}		

FIG.8

Syntax	Number of Bits	Mnemonic
play_protect() {		
play_protect_length	32	uimbsf
volume_play_protect_flag	2	bslbf
reserved	2	bslbf
password_enable_flag	1	bslbf
reserved	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_play_protect	128	bslbf
reserved	8	bslbf
play_protect_set_time()	56	bcd
reserved	8	bslbf
play_protect_expiration_time()	56	bcd
number_of_times	16	uimbsf
}		

FIG.9

Syntax	Number of Bits	Mnemonic
recording_timer() {		
recording_timer_length		
recording_timer_flag		
number_of_entry		
for(i=0;i<number_of_entry;i++){		
date_and_time		
channel		
program		
:		
}		
}		

FIG. 10

Syntax	Number of Bits	Mnemonic
text_block() {		
text_block_length	32	uimsbf
number_of_language_sets	8	uimsbf
number_of_text_items	16	uimsbf
for(i=0;i<number_of_language_sets;i++) {		
language_set()		
}		
for(i=0;i<number_of_text_items;i++) {		
text_item()		
}		
}		

FIG. 11

Syntax	Number of Bits	Mnemonic
language_set() {		
reserved	8	bslbf
language_code	24	bslbf
character_set_type	8	bslbf
number_of_language_set_names	8	uimsbf
for(i=0;i<number_of_language_set_names;i++) {		
character_set_type_for_name	8	bslbf
language_set_name_length	8	uimsbf
language_set_name 8*language_set_name_length		bslbf
}		
}		

FIG. 12

Syntax	Number of Bits	Mnemonic
text_item() {		
text_item_length	16	uimsbf
text_item_id	16	uimsbf
text_item_sub_id	16	uimsbf
flags	8	bslbf
number_of_used_language_sets	8	uimsbf
//loop for each language set		
for(i=0;i<number_of_used_language_sets;i++) {		
language_set_id	8	uimsbf
reserved	4	bslbf
text_string_length	16	uimsbf
text_string	8*text_string_length	bslbf
bitmap()		
}		
stuffing_bytes	8*n	bslbf
}		

FIG. 13

Syntax	Number of Bits	Mnemonic
ALBUM. STR{		
file_type_id	8*16	char[16]
album()		
text_block()		
}		

FIG. 14

Syntax	Number of Bits	Mnemonic
album() {		
album_length	32	uimsbf
reserved	6	bslbf
volume_status	1	bslbf
if (volume_status== "1b") {		
chief_volume_flag	1	bslbf
} else {		
reserved	1	"0"
}		
if (volume_status== "1b") {		
if (chief_volume_flag== "1b") {		
reserved	6	bslbf
album_type	2	bslbf
albm_id	128	bslbf
number_of_discs_in_album	16	uimsbf
number_of_volumes_in_album	16	uimsbf
for (i=0;i<number_of_volumes_in_album;i++) {		
disc_id_for_album_member	128	bslbf
volume_id_for_album_member	128	bslbf
title_offset_number	16	uimsbf
}		
reserved_for_program_bind	8	bslbf
number_of_program_binds	8	uimsbf
for (i=0;i<number_of_program_binds;i++) {		
number_of_program_in_this_program_bind	16	uimsbf
for(i=0;i<number_of_programs_in_this_program_bind;i++){		
disc_id_for_program_bind_member	128	uimsbf
volume_id_for_program_bind_member	128	uimsbf
program_number	16	uimsbf
}		
}		
} else {		
//chief_volume_flag== "0b"		
chief_disc_id	128	uimsbf
chief_volume_id	128	uimsbf
(album_id	128	bslbf
}		
}		
}		

FIG. 15

Syntax	Number of Bits	Mnemonic
TITLE_###. VDR{ file_type_id title_info() text_block() }	8*16	char [16]

FIG. 16

Syntax	Number of Bits	Mnemonic
title_info() { title_info_length flags_for_title cgit_file_id title_start_chunk_group_time_stamp title_end_chunk_group_time_stamp title_playback_time() reserved number_of_marks for (i=0;i<number_of_marks;i++) { reserved mark_type mark_chunk_group_time_stamp } stuffing_bytes }	 32 32 16 64 64 32 32 16 4 4 64 8*n	 uimbsf bslbf uimbsf uimbsf uimbsf bcd bslbf uimbsf bslbf bslbf uimbsf bslbf

FIG. 17

Syntax	Number of Bits	Mnemonic
PROGRAM_\$\$\$ PGI { file_type_id program() text_block() }	8*16	char[16]

FIG. 18

Syntax	Number of Bits	Mnemonic
program() {		
program_length	32	uimsbf
flags_for_program	32	bslbf
program_status	4	bslbf
program_playback_time()	32	bslbf
reserved	32	bslbf
number_of_play_sequences	16	uimsbf
for (j=0; j<number_of_play_sequence; j++) {		
number_of_play_lists	16	uimsbf
for (k=0; k<number_of_play_lists; k++) {		
play_list_start_time_stamp_offset	64	uimsbf
play_list(k)		
}		
}		
stuffing_bytes	8*n	bslbf
}		

FIG. 19

Syntax	Number of Bits	Mnemonic
play_list() {		
//playback sequence of play items in this play list		
number_of_play_items	16	uimsbf
for (k=0; k<number_of_play_items; k++) {		
play_item_number	16	uimsbf
reserved	31	bslbf
seamless_connection_flag	1	bslbf
}		
//play_item_table		
for (PIN=1; PIN<=number_of_play_items_in_program; PIN++) {		
play_item()		
}		
}		

FIG. 20

Syntax	Number of Bits	Mnemonic
play_item() {		
play_item_length	32	uimsbf
play_item_type	8	bslbf
play_mode	8	bslbf
total_playback_time()	32	bcd
menu_item_number	16	uimsbf
return_item_number	16	uimsbf
next_item_number	16	uimsbf
prev_item_number	16	uimsbf
if(play_item_type= "0000b") {		
//play item for one "cut"		
title_number	16	uimsbf
//IN point		
item_start_time_stamp	64	uimsbf
//OUT point		
item_end_time_stamp	64	uimsbf
}		
}		

FIG. 21

Syntax	Number of Bits	Mnemonic
CHUNKGROUP_###. CGIT{		
file_type_id	8*16	char[16]
chunkgroup_time_base_flags	32	bslbf
chunkgroup_time_base_offset	64	uimsbf
chunk_connection_info()		
text_block()		
}		

F I G . 2 2

Syntax	Number of Bits	Mnemonic
chunk_connection_info() {		
chunk_connection_info_length	32	uimsbf
reserved	16	bslbf
number_of_chunks	16	uimsbf
chunk_sync_play_flag	8	bslbf
// chunk info file list		
for(i=0; i<number_of_chunks;i++){		
chunk_arrangement_info()		
}		
}		

F I G . 2 3

Syntax	Number of Bits	Mnemonic
chunk_arrangement_info() {		
chunk_arrangement_info_length	32	uimsbf
chunk_info_file_id	16	bslbf
reserved	5	bslbf
chunk_switch_stream_id	16	bslbf
presentation_start_cg_time_count	64	uimsbf
presentation_end_cg_time_count	64	uimsbf
reserved	4	bslbf
chunk_time_count_type	4	bslbf
number_of_start_original_time_count_extension	8	uimsbf
number_of_end_original_time_count_extension	8	uimsbf
// presentation start position and time		
presentation_start_original_time_count	64	uimsbf
presentation_end_original_time_count	64	uimsbf
for(i=0;j<number_of_start_original_time_count_extension;j++){		
tc_ext_attributes	16	bslbf
start_original_time_count_extension	64	uimsbf
}		
// presentation end position and time		
for(k=0;k<number_of_end_original_time_count_extension;k++){		
tc_ext_attributes	16	bslbf
end_original_time_count_extension	64	uimsbf
}		
transition_info()		
}		

FIG. 24

Syntax	Number of Bits	Mnemonic
CHUNK_%%%.ABST(file_type_id info_type reserved cognizant_recording_indicator //stream_info() if (info_type == "MPEG2_system_TS") { number_of_programs } else { number_of_programs } for(i=0;i<number_of_programs;i++){ number_of_streams for (i=0;i<number_of_streams;i++) { stream_identifier //slot type information reserved slot_unit_type if (slot_unit_type=="time_stamp") { slot_time_length } else { reserved } number_of_slots number_of_thinned_out_slots //stream attribute bitstream_attribute() } //loop of slot information for (i=0;i<number_of_streams;i++) { slot_info() } } text_block())	8*16 4 3 1 8 8 8 16 4 4 32 32 32 8	char[16] bslbf bslbf bslbf uimbsf '0000 0001' uimbsf bslbf bslbf uimbsf bslbf uimbsf uimbsf

FIG. 25A

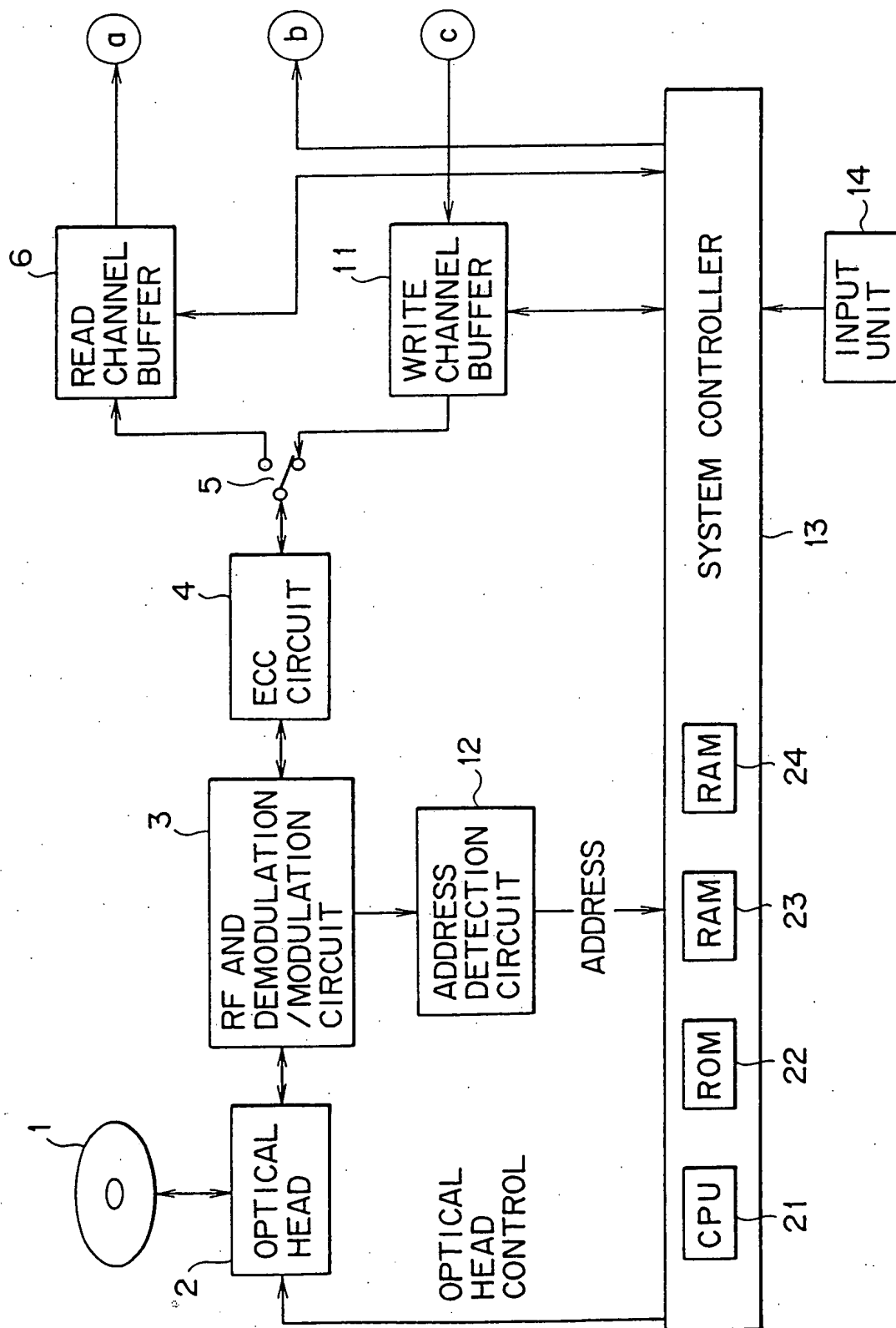


FIG. 25B

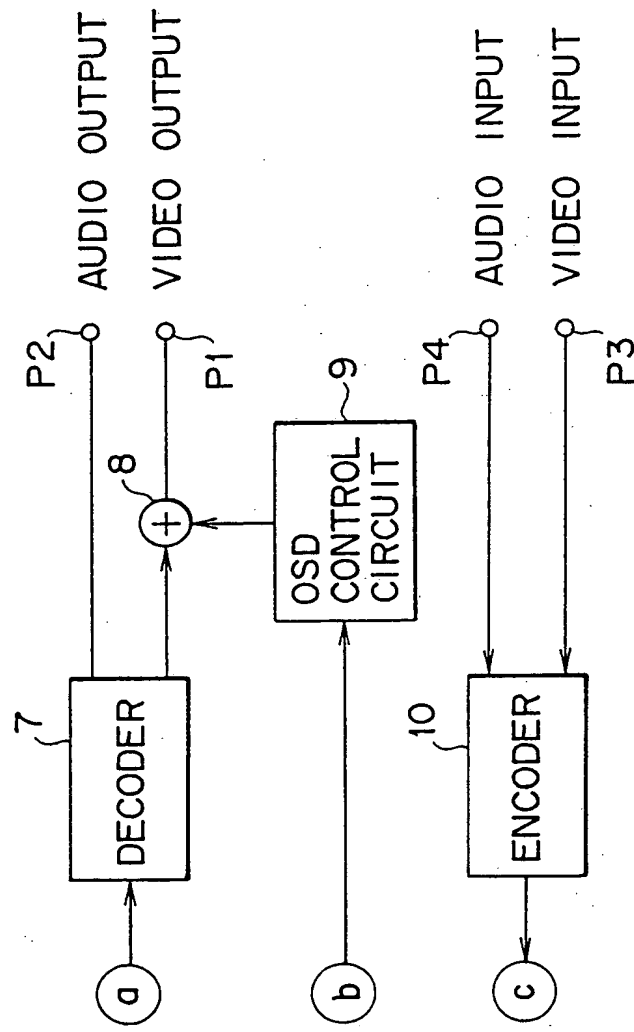


FIG. 25

FIG. 25A	FIG. 25B
----------	----------

FIG. 26

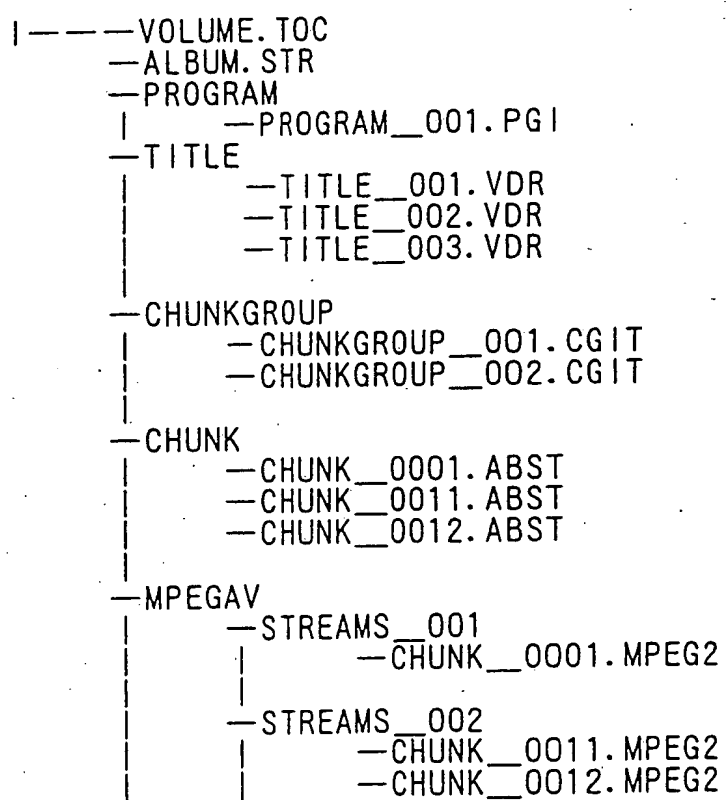


FIG. 27

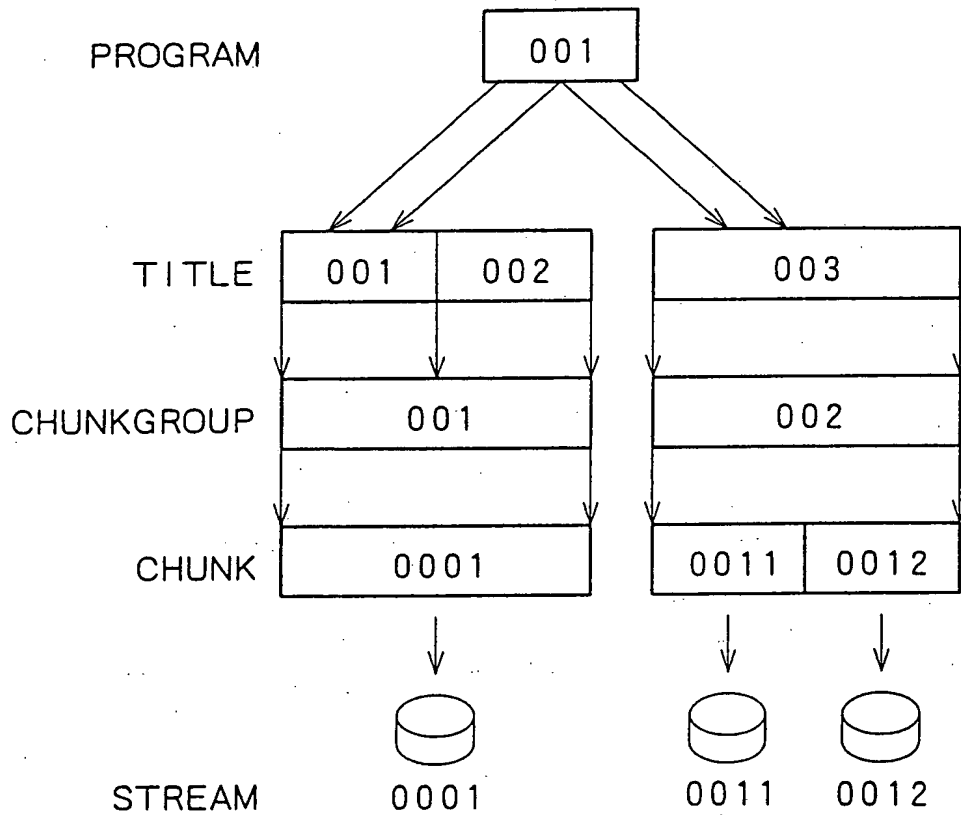


FIG. 28

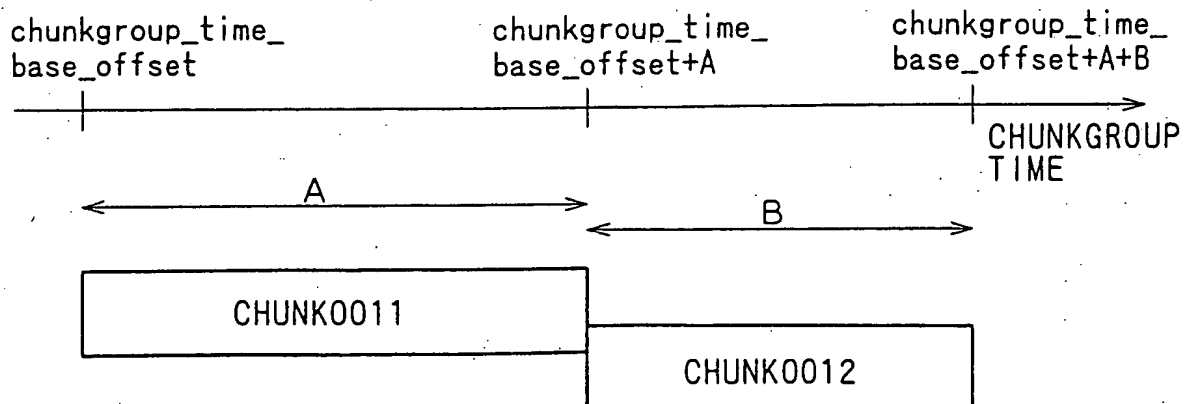


FIG. 29

```

/-----MPEGAV
|         -STREAMS_003
|         |         -CHUNK_0031.MPEG2

```

FIG. 30

```

/---VOLUME.TOC
-ALBUM.STR
-PROGRAM
|   -PROGRAM_001.PGI
-TITLE
|   -TITLE_001.VDR
|   -TITLE_002.VDR
|   -TITLE_003.VDR
|   -TITLE_004.VDR*
-CHUNKGROUP
|   -CHUNK_001.CGIT
|   -CHUNK_002.CGIT
|   -CHUNK_003.CGIT*
-CHUNK
|   -CHUNK_0001.ABST
|   -CHUNK_0011.ABST
|   -CHUNK_0012.ABST
|   -CHUNK_0031.ABST*
-MPEGAV
|   -STREAMS_001
|       |   -CHUNK_0001.MPEG2
|       |
|   -STREAMS_002
|       |   -CHUNK_0011.MPEG2
|       |   -CHUNK_0012.MPEG2
|       |
|   -STREAMS_003*
|       |   -CHUNK_0031.MPEG2*
|

```

FIG. 31

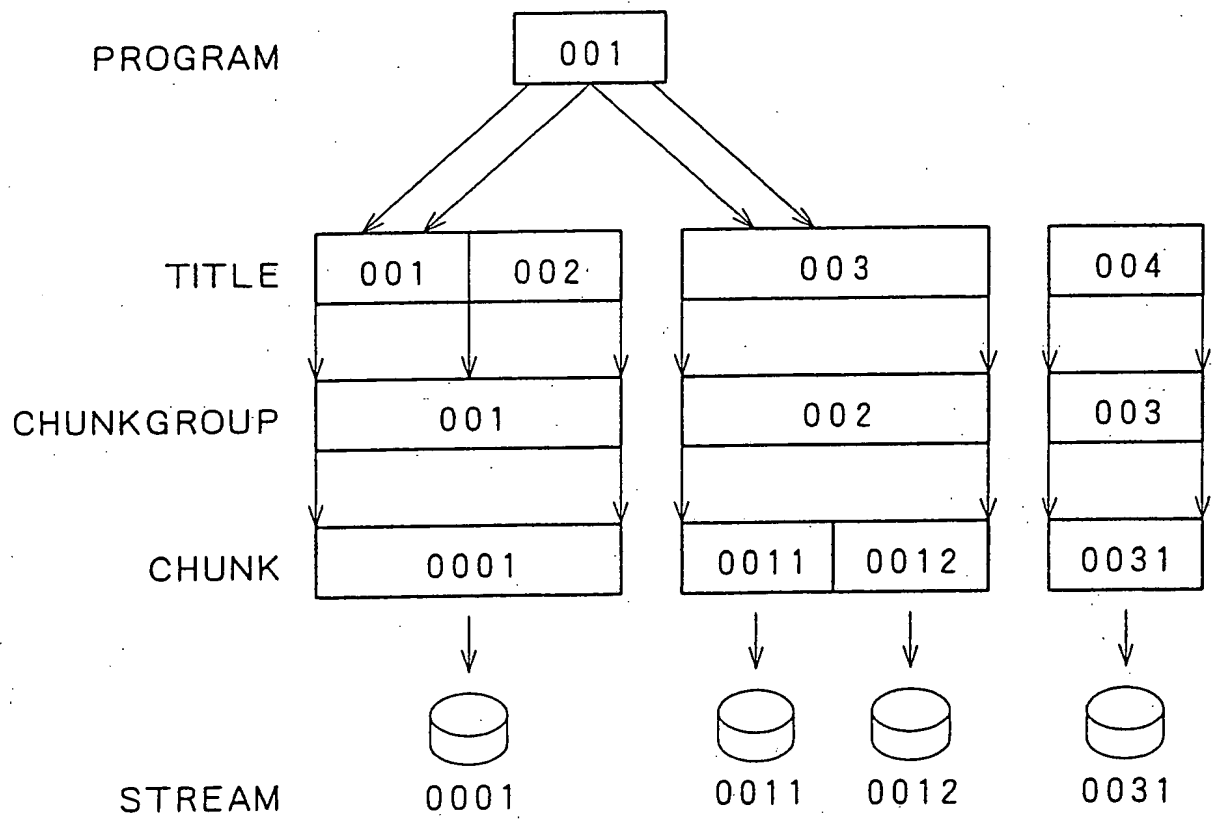


FIG. 32

```
/-----MPEGAV
|         -STREAMS_002
|         |         -CHUNK_0031.MPEG2
```

FIG. 33

```
/---VOLUME.TOC
-ALBUM.STR
-PROGRAM
|   -PROGRAM_001.PGI
-TITLE
|   -TITLE_001.VDR
|   -TITLE_002.VDR
|   -TITLE_003.VDR
|   -TITLE_004.VDR*
-CHUNKGROUP
|   -CHUNKGROUP_001.CGIT
|   -CHUNKGROUP_002.CGIT
-CHUNK
|   -CHUNK_0001.ABST
|   -CHUNK_0011.ABST
|   -CHUNK_0012.ABST
|   -CHUNK_0031.ABST*
-MPEGAV
|   -STREAMS_001
|       -CHUNK_0001.MPEG2
|   -STREAMS_002
|       -CHUNK_0011.MPEG2
|       -CHUNK_0012.MPEG2
|       -CHUNK_0031.MPEG2*
```

FIG. 34

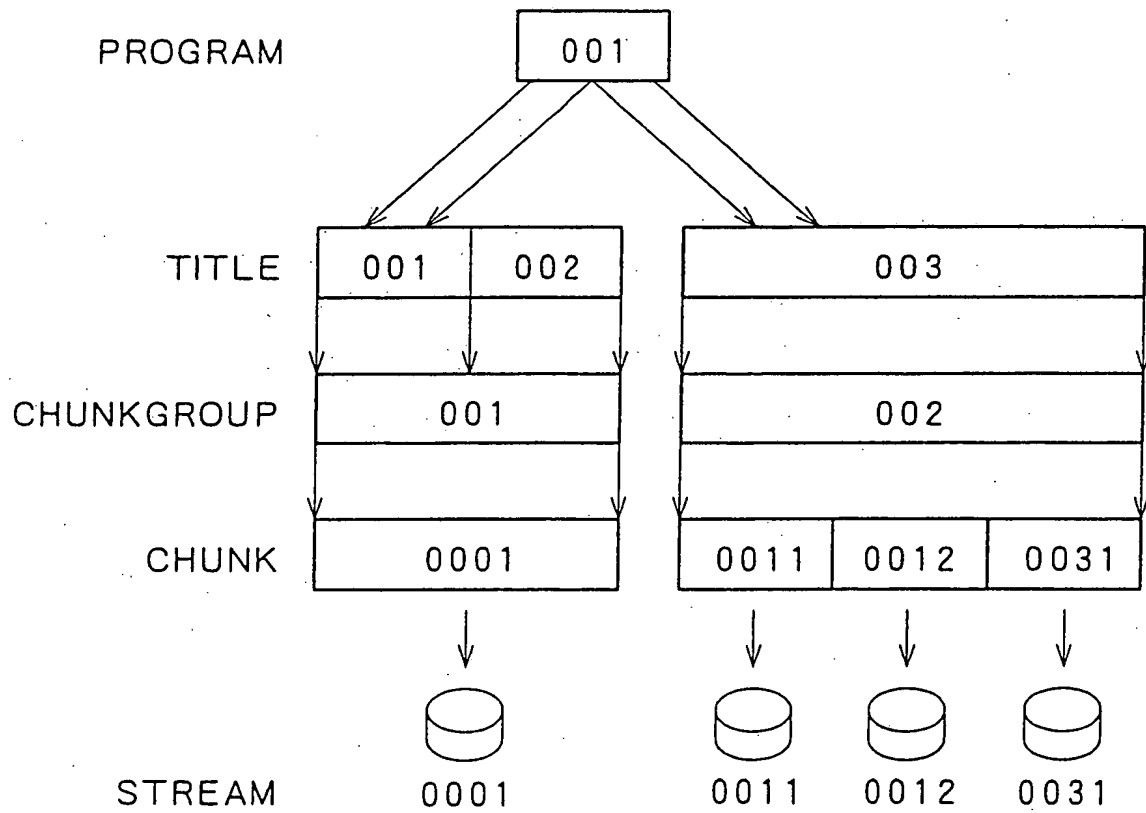


FIG. 35

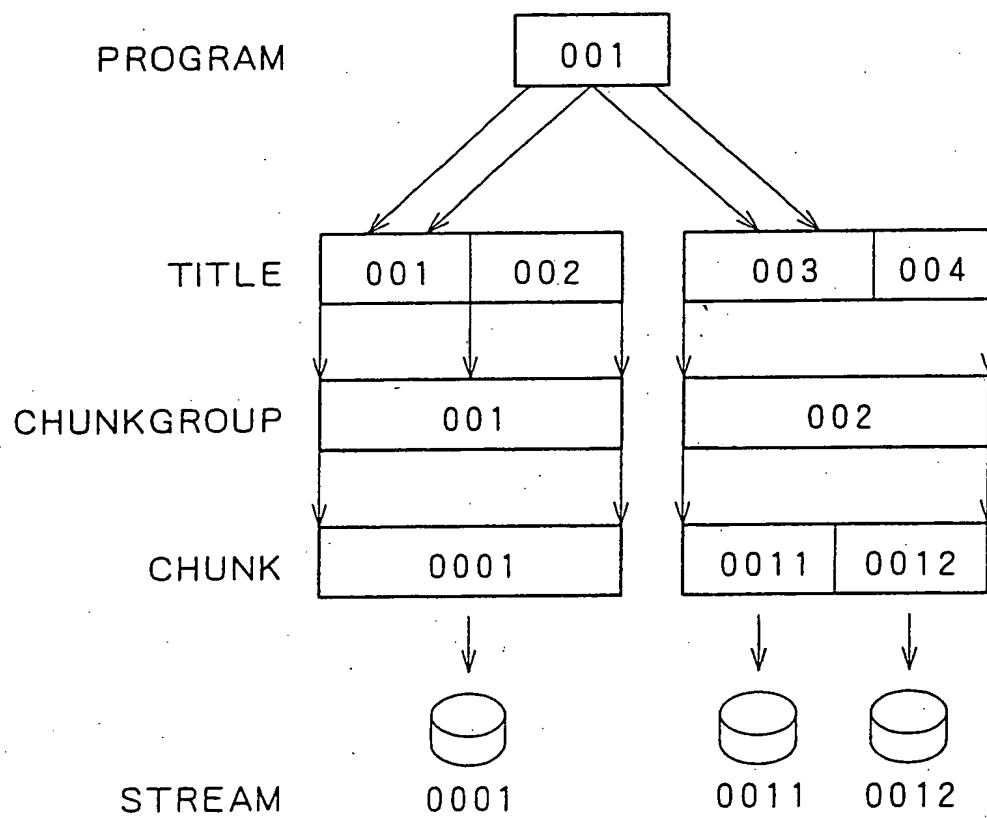


FIG. 36A

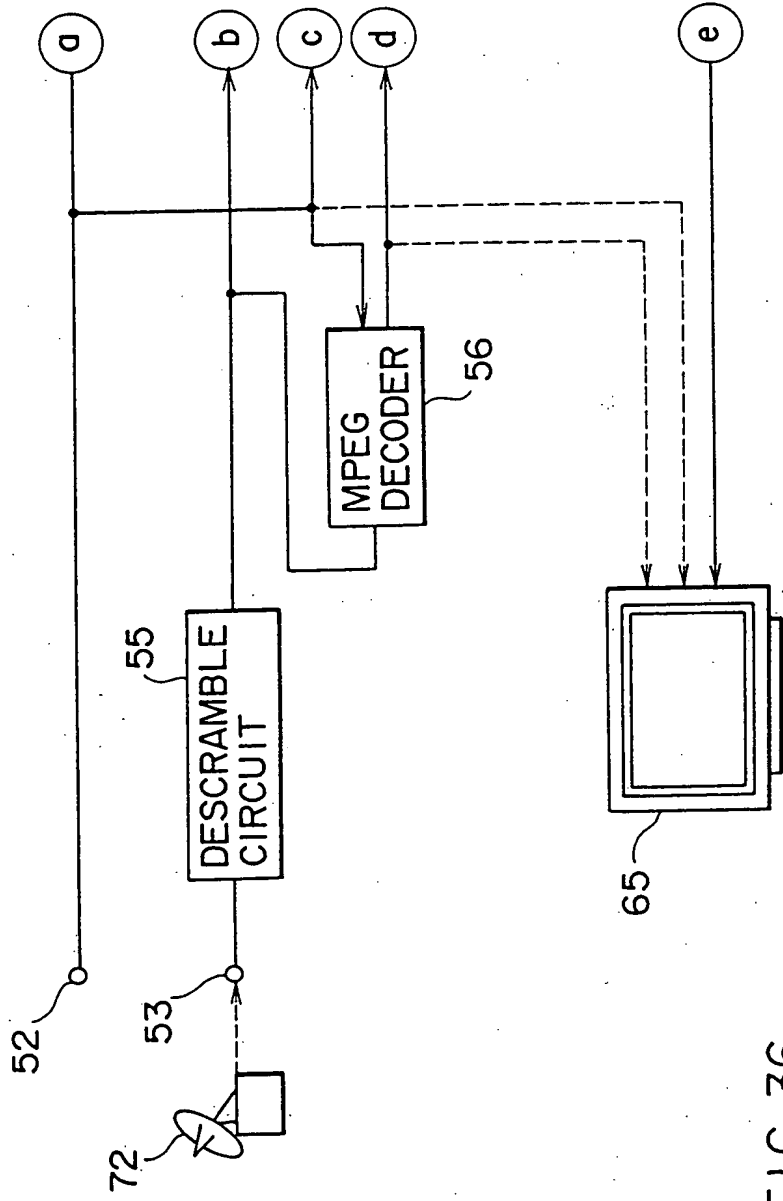


FIG. 36

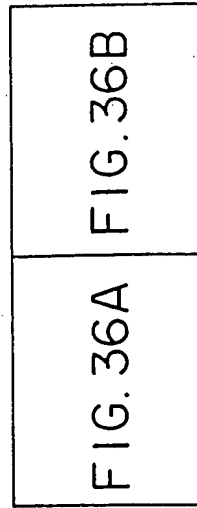


FIG. 36B

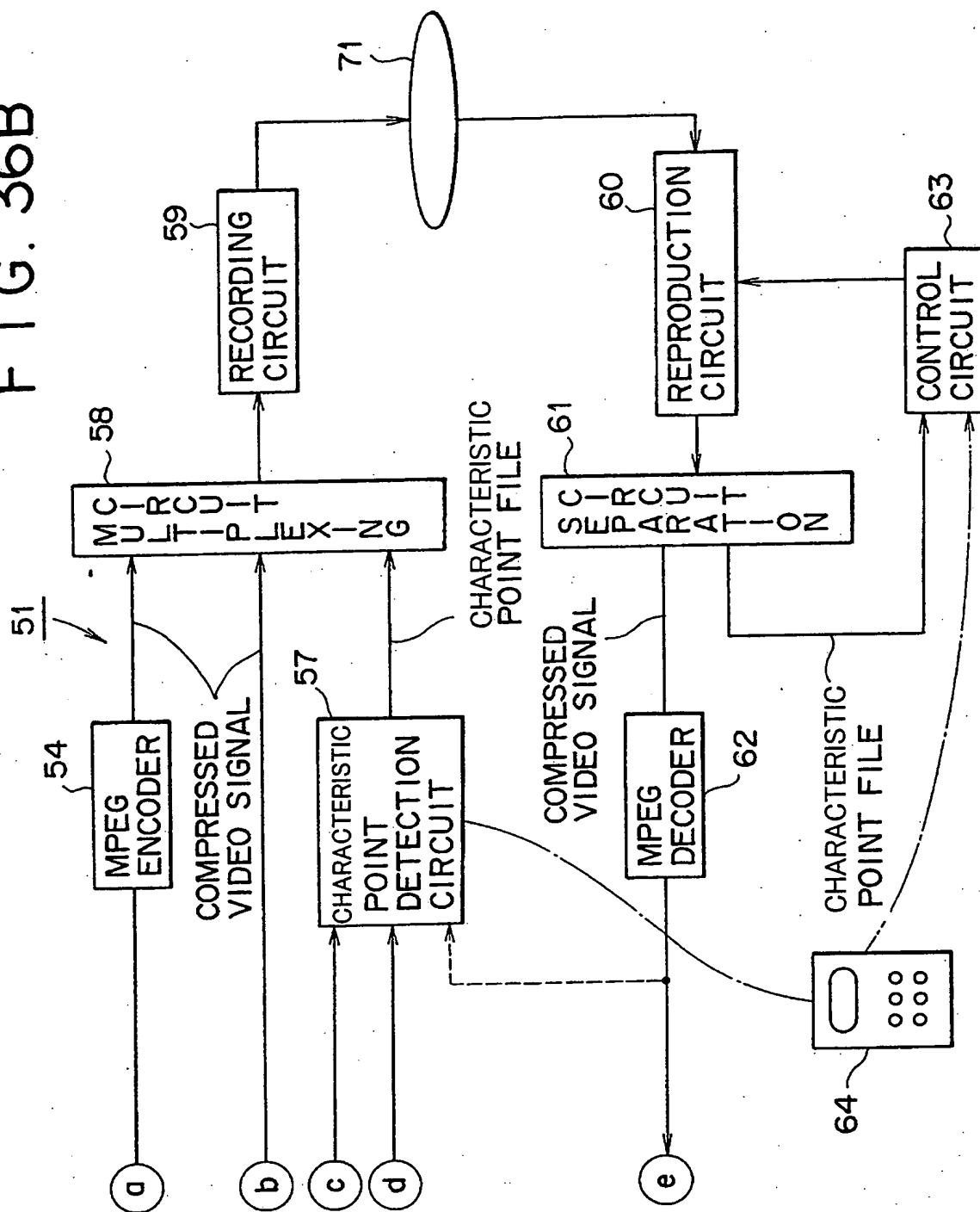


FIG. 37A

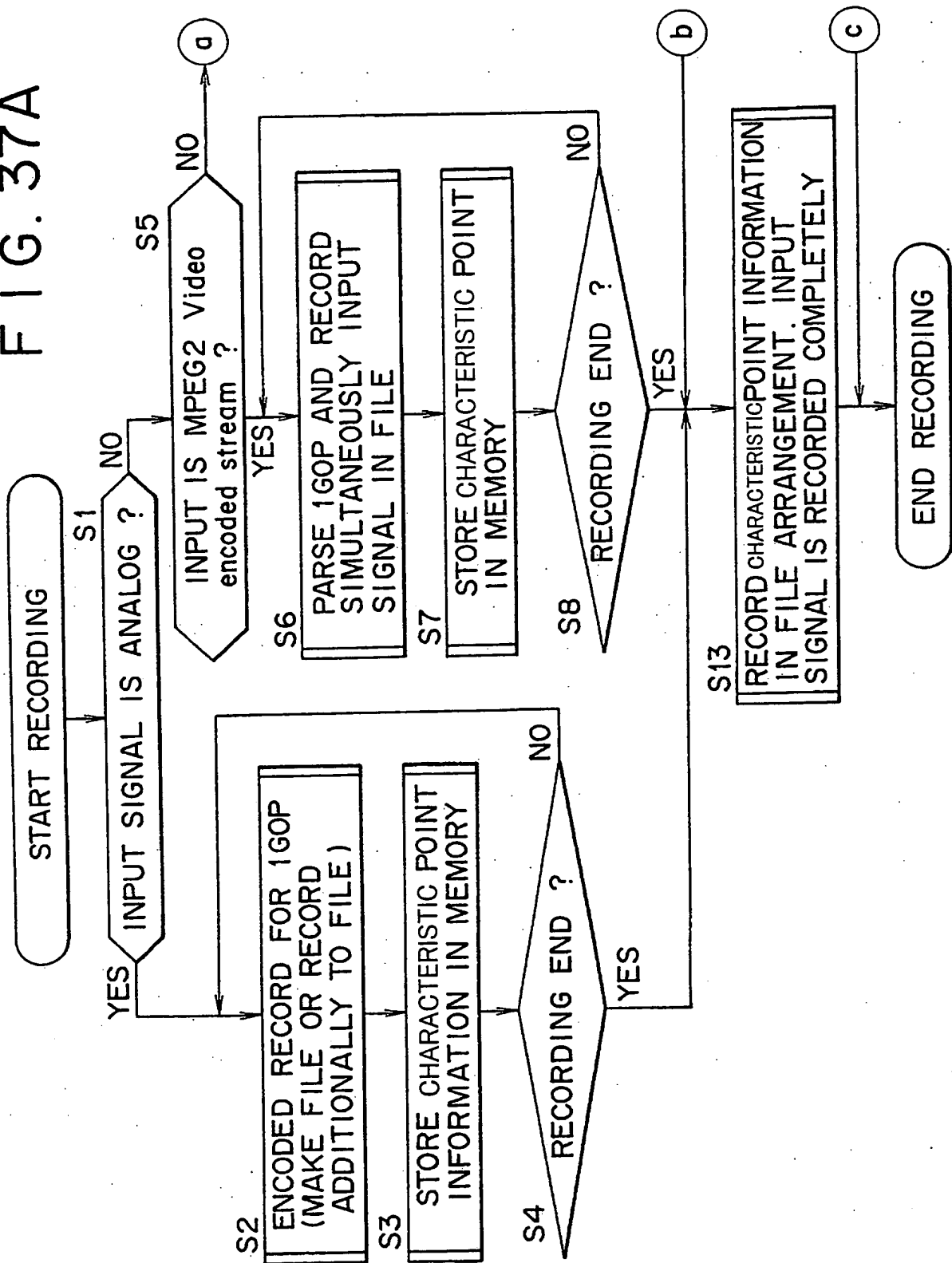


FIG. 37B

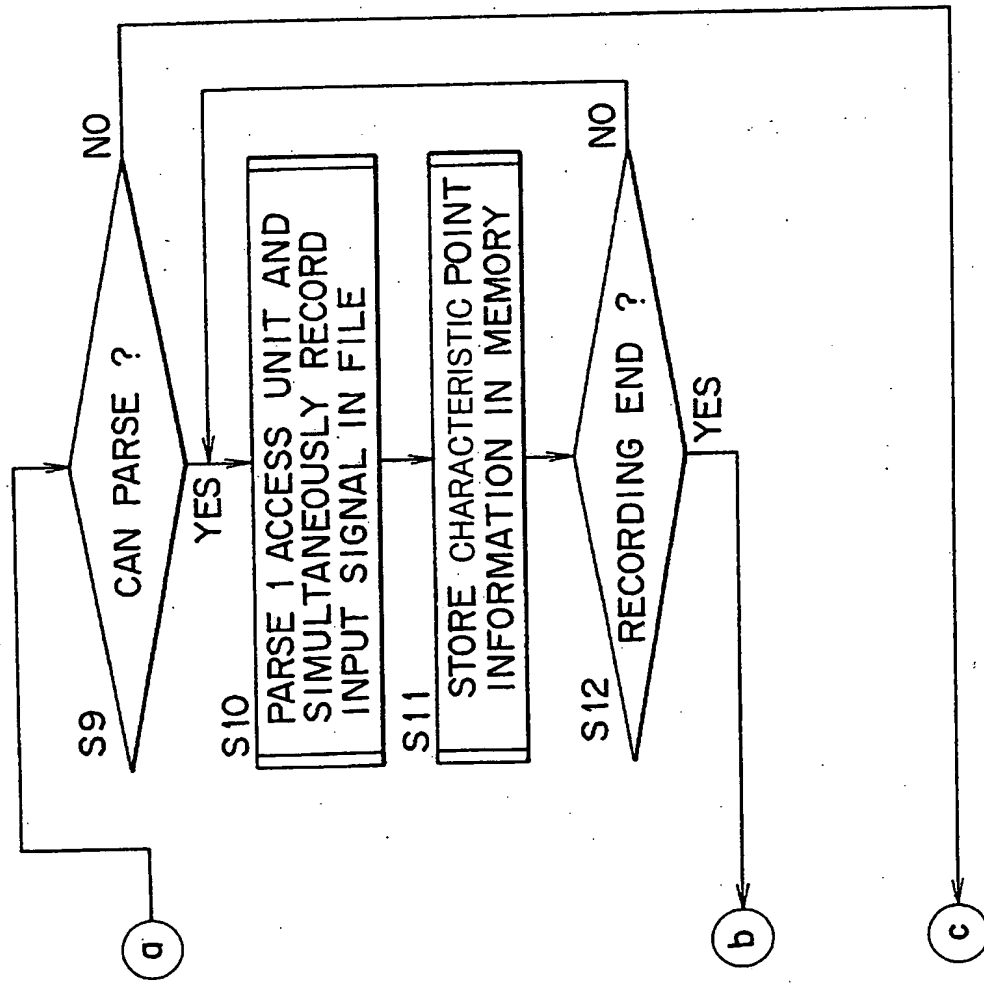


FIG. 37

FIG. 37A	FIG. 37B
----------	----------

FIG.38

field name	value
file_type_id	"STRM_INF_FILE_>"

FIG.39

info_type	Meaning
0000	MPEG2_System_PS
0001	MPEG2_System_TS
0010	MPEG2_System_PES
0011	MPEG1_System_stream
0100 . . 0111	reserved
1000	Consumer_DVC
1001 . . 1111	reserved

FIG.40

cognizant_recoding_indicator	Meaning
0b	This chunk was recorded by noncognizant device
1b	This chunk was recorded by cognizant device

FIG.41

slot_unit_type	Meaning
0000b	'time_stamp':time stamp value
0001b	'GOP' : one GOP(Group of pictures)
0010b	'audio_frame' : one audio frame
0011b . . 1111b	reserved

FIG.42

Syntax	Number of Bits	Mnemonic
<pre> bitstream_attribute() { bitstream_attribute_id bitstream_attribute_length reserved attribute_type switch(attribute_type) { case video: video_attribute() break; case audio: audio_attribute() break; default: break; } </pre>	<p>8*16</p> <p>32</p> <p>4</p> <p>4</p>	<p>char [16]</p> <p>uimbsf</p> <p>bslbf</p> <p>bslbf</p>

FIG.43

field name	value
bitstream_attribute_id	"STRM_ATTRBT_00->"

FIG.44

attribute_type	Meaning
0000b	video
0001b	audio
0010b	reserved for graphics
0011b	reserved for PBC
0100b . . 1111b	reserved

F I G . 4 5

Syntax	Number of Bits	Mnemonic
video_attribute() {		
input_video_source	4	bslbf
video_compression_mode	4	bslbf
picture_rate	4	bslbf
picture_scan_type	4	bslbf
vertical_lines	4	bslbf
horizontal_active_pixels	16	uimbsbf
aspect_ratio	4	bslbf
pixel_ratio	2	bslbf
CC_existence	2	bslbf
recording_mode	4	bslbf
copyright_information	64	bslbf
}		

F I G . 4 6

input_video_source	Meaning
0000b	unknown
0001b	inapplicable
0010b	analog component input
0011b	analog composit input
0100b	analog Y/C separated input
0101b	local digital bus (internal IRD etc.)
0110b	IEEEE 1394
0111b	SDI (Serial Digital Interface)
1000b . . 1111b	reserved

FIG.47

video_compression_mode	Meaning
0000b	unknown
0001b	inapplicable
0010b	MPEG1 Video
0011b	MPEG2 Video
0100b	DV
0101b . . 1111b	reserved

FIG.48

picture_rate	Meaning
0000b	unknown
0001b	24000/1001 (Hz)
0010b	24
0011b	25
0100b	30000/1001
0101b	30
0110b	50
0111b	60000/1001
1000b	60
1001b . . 1111b	reserved

FIG.49

picture_scan_type	Meaning
0000b	unknown
0001b	inapplicable
0010b	progressive
0011b	interlace
0100b . . 1111b	reserved

FIG.50

vertical_lines	Meaning: number of active vertical lines (total number of vertical lines)
0000b	unknown
0001b	inapplicable
0010b	480 (525)
0011b	720 (750)
0100b	1035 (1125)
0101b	1080 (1125)
0110b . . 1111b	reserved

FIG.51

aspect_ratio	Meaning
0000b	unknown
0001b	inapplicable
0010b	4 : 3
0011b	16 : 9
0100b	2.21 : 1
1001b	1 : 1
0110b . . 1111b	reserved

FIG.52

pixel_ratio	Meaning
0000b	unknown
0001b	inapplicable
0010b	1 : 1
0011b	1 : 1.125
0100b	reserved
0101b . . 1111b	reserved

FIG.53

CC_existence	Meaning
0000b	unknown
0001b	inapplicable
0010b	exist
0011b	not exist
0100b	reserved
0101b . . 1111b	reserved

FIG.54

recording_mode	Meaning
0000b	unknown
0001b	inapplicable
0010b	Standard Play
0011b	Long Play
0100b	reserved
0101b . . 1111b	reserved

FIG. 55

Syntax	Number of Bits	Mnemonic
audio_attribute() {		
reserved	4	bslbf
number_of_audio_streams	4	uimsbf
for (j=0;j<number_of_audio_streams;j++) {		
stream_id	8	bslbf
sub_stream_id	8	bslbf
reserved	8	bslbf
language_code	24	bslbf
input_source	4	bslbf
audio_coding_mode	4	bslbf
bitrate	8	bslbf
q_bit	4	bslbf
fs	4	bslbf
reserved	7	bslbf
emphasis	1	bslbf
number_of_channels	8	uimsbf
for (k=0;k<number_of_channels;k++) {		
reserved	4	bslbf
status_of_this_channel	4	bslbf
channel_assignment	8	bslbf
}		
(Dynamic_range_control)		
karaoke()		
}		
}		

FIG. 56

input_source	Meaning
0000b	unknown
0001b	inapplicable
0010b	anlog input
0011b	reserved
0100b	internal IRD
1001b	IEEE 1394
0110b	IEC 958
0111b . . 1111b	reserved

FIG.57

audio_coding_mode	Meaning
0000b	unknown
0001b	inapplicable
0010b	LPCM
0011b	AC-3
0100b	MPEG Audio
1001b	ATRAC
0101b . . 1111b	reserved

FIG.58

bitrate	Meaning
0000 0000b	variable bitrate
0000 0001b	inapplicable
0000 0010b ..1111 1111b	reserved

FIG.59

q_bit	Meaning
0000b	unknown
0001b	inapplicable
0010b	8
0011b	12
0100b	16
1001b	20
0110b	24
0111b . . 1111b	reserved

FIG.60

fs	Meaning
0000b	unknown
0001b	inapplicable
0010b	8 kHz
0011b	16 kHz
0100b	32 kHz
1001b	44.1 kHz
0110b	48 kHz
0111b	96 kHz
1000b . . . 1111b	reserved

FIG.61

emphasis	Meaning
0b	emphasis off
1b	emphasis on

FIG.62

Syntax	Number of Bits	Mnemonic
slot_info() {		
slot_info_id	8*16	char[16]
slot_info_length	32	uimbsf
switch(slot_unit_type) {		
case 'GOP':		
for (i=0;i<number_of_slots;i++) {		
slot_info_for_one_GOP()		
}		
break;		
case 'audio_frame':		
for (i=0;i<number_of_slots;i++) {		
slot_info_for_one_audio_frame()		
}		
break;		
case 'time_stamp':		
for (i=0;i<number_of_slots;i++) {		
slot_info_for_one_time_slot()		
}		
break;		
}		
}		

FIG.63

field name	value
slot_info_id	"SLOT_INF_00000->"

FIG. 64

Syntax	Number of Bits	Mnemonic
slot_info_for_one_GOP() {		
slot_length	24	uimsbf
//specific info		
switch (info_type) {		
case MPEG1_System_stream:		
sequence_header_start_offset	24	uimsbf
packet_start_offset	24	uimsbf
pack_start_offset	24	uimsbf
break;		
case MPEG2_System_PS:		
sequence_header_start_offset	24	uimsbf
PES_packet_start_offset	24	uimsbf
pack_start_offset	24	uimsbf
break;		
case MPEG2_System_TS:		uimsbf
sequence_header_start_offset	24	uimsbf
TS_packet_start_offset	16	uimsbf
PES_packet_start_offset	24	uimsbf
TS_packet_start_offset2	16	uimsbf
break;		
case MPEG2_System_PES:	24	uimsbf
sequence_header_start_offset	24	uimsbf
PES_packet_start_offset		
break;		
}		
GOPH_existence_flag	1	bslbf
first_presented_picture_structure	2	bslbf
copy_closed_GOP	1	bslbf
copy_broken_link	1	bslbf
reserved	2	bslbf
time_stamp_of_first_picture	33	uimsbf
GOP_status	6	bslbf
picture_count_type	2	bslbf
number_of_pictures	16	uimsbf
encode_info()	32	bslbf
buffer_occupancy()	32	bslbf
camera_info()	16	bslbf
reserved	16	bslbf
}		

FIG.65

first_presented_picture_structure	Meaning
00b	top field
01b	bottom field
10b . . 11b	reserved

FIG.66

picture_count_type	Meaning
00b	field
01b	field
10b . . 11b	reserved

FIG.67

Syntax	Number of Bits	Mnemonic
slot_info_for_one_audio_frame() {		
AAU_storage_length	32	uimbsf
//specific info		
AAU_start_byte_position	16	uimbsf
flags	7	bslbf
encode_info()	32	bslbf
camera_info()	16	bslbf
}		

FIG.68

Syntax	Number of Bits	Mnemonic
slot_info_for_one_time_slot() {		
}		